REPORT DOCUMENTATION PAGE

The public reporting burden for this collection of information is estimated to average 1 hour per response, including gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments of information, including suggestions for reducing the burden, to Department of Defense, Washington Headqu (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be awa subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

AFRL-SR-BL-TR-01-
0219

1. REPORT DATE (DD-MM-YYYY) 03/12/2001	2. REPORT TYPE FINAL TECHNICAL REPORT	3. DATES COVERED (From - To) DEC 1999 -JAN 2001

4. TITLE AND SUBTITLE
"An Implicit, Conservative Multi-Temperature MHD Algorithm"

5a. CONTRACT NUMBER
F49620-00-1-0102

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)
SHUMLAK, URI

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Washington
Grant & Contract Services
3935 University Way NE
Seattle, WA 98105-6613

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research (ONR)
1107 NE 45th Street, Suite 350
Seattle, WA 98105-4631

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
unclassified/unlimited

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
NOTICE OF TRANSMITTAL DTIC. THIS TECHNICAL REPORT
HAS BEEN REVIEWED AND IS APPROVED FOR PUBLIC RELEASE
LAW AFR 190-12. DISTRIBUTION IS UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
A algorithm to accurately simulate plasmas with constituent species at multiple temperature using the MHD model was developed. The algorithm was based on a Roe-type approximate Riemann solver. The algorithm was implemented in a code to model the time-dependent, three-dimensional, arbitrary-geometry MHD model which includes viscous and resistive effects. A time-dependent ionization model was added which self-consistently calculates the ionization fraction of the fluid. Energy loss mechanisms were added for the constituent fluid components (neutrals, ions, and electrons). The algorithm was implemented on parallel supercomputers to allow the detailed modeling of realistic plasmas in complex three-dimensional geometries.

15. SUBJECT TERMS
MHD Model, Roe-type, Riemann solver, three-demensional, arbitrary-geometry, time-dependent

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Uri Shumlak
U	U	U	UU	23	19b. TELEPHONE NUMBER (Include area code) 206-616-1986

An Implicit, Conservative Multi-Temperature MHD Algorithm

Contract Number: F49620-00-1-0102

U. Shumlak
Department of Aeronautics and Astronautics, Box 352250
University of Washington, Seattle, WA 98195-2250

Abstract

A algorithm to accurately simulate plasmas with constituent species at multiple temperatures using the MHD model was developed. The algorithm was based on a Roe-type approximate Riemann solver. The algorithm was implemented in a code to model the time-dependent, three-dimensional, arbitrary-geometry MHD model which includes viscous and resistive effects. A time-dependent ionization model was added which self-consistently calculates the ionization fraction of the fluid. Energy loss mechanisms were added for the constituent fluid components (neutrals, ions, and electrons). The algorithm was implemented on parallel supercomputers to allow the detailed modeling of realistic plasmas in complex three-dimensional geometries.

20010404 098

Contents

1 Executive Summary **1**

2 Project Description **1**
2.1 Research Objectives 2
2.2 Technical Description 2
2.3 Multiple Temperature Evolution 4
2.4 Conservative Algorithm 5
2.5 Implicit Formulation 7
2.6 Approximate Riemann Solver 9
2.7 Finite Volume Grid and Parallel Implementation 10

3 Project Implementation and Results **12**
3.1 Finite Volume Improvements 12
3.2 Implicit Formulation and Numerical Flux Jacobian Calculations 14
3.3 Time Dependent Ionization and Multiple Temperature Effects 17
3.4 Parallel Computer Performance 18

4 Professional Interactions **20**
4.1 Project Personnel 20
4.2 Collaborations 21
4.2.1 Air Force Research Laboratory 21
4.2.2 Sandia National Laboratories 21
4.2.3 University of Washington 21
4.3 Publications 22

5 Conclusions **22**

List of Figures

1 Schematic of the arbitrary shaped three-dimensional finite volume cell used by the algorithm. 11

2 Schematic of the arbitrary shaped three-dimensional finite volume cell used by the algorithm. 11

3 A three-dimensional simulation of a one-dimensional shock tube showing the density and transverse velocity. The three gas dynamic waves can be seen in the density plot. The transverse velocity should be zero. 13

4 A three-dimensional simulation of a one-dimensional shock tube showing the density and transverse velocity. The three gas dynamic waves can be seen in the density plot. The transverse velocity should be zero. 14

5 Convergence history using the LU-SGS method to invert the implicit operator. n is the number of physical time iterations, and m is the number of LU-SGS pseudo time subiterations. 15

6 Convergence history using the SGS method to invert the implicit operator. n is the number of physical time iterations, m is the number of pseudo time subiterations, and sgs is the number of iterations of the SGS method. 16

7 Mach 6 flow impinging on a hemispherical body. The upper plot (a) shows the density contours. The lower plot (b) shows the ionization fraction. Notice the increased ionization at the stagnation point. 17

8 (a) Strip decomposition and (b) patch decomposition of a 2-D domain. . 18

9 Convergence history using the SGS method to invert the implicit operator on a parallel computer. The results from a serial computer are plotted for comparison. n is the number of physical time iterations, m is the number of pseudo time subiterations, and sgs is the number of iterations of the SGS method. 19

10 Parallel speedup for a three-dimensional grid using domain decomposition on a cluster of DEC Alpha workstations. The grid is scaled with the number of processors, so the grid size per processor is constant. The ideal speedup is unity. 20

11 Parallel speedup for a three-dimensional grid using domain decomposition on IBM SP2 parallel supercomputer. The grid is scaled with the number of processors, so the grid size per processor is constant. The ideal speedup is unity. 21

1 Executive Summary

The primary objective of this project is to develop a novel algorithm to accurately simulate realistic plasmas with constituent species at multiple temperatures using the MHD model. A viable time-dependent, three-dimensional MHD code will provide a valuable tool for the design and testing of plasma related technologies that are important to the Air Force and industry, such as portable pulsed power, high power microwave devices, hypersonic drag reduction, advanced plasma thrusters for space propulsion, nuclear weapons effects simulations, radiation production for counter proliferation, and fusion for power generation. Implementing the algorithm on parallel supercomputers will allow the detailed modeling of realistic plasmas in complex three-dimensional geometries.

Current MHD codes are limited to simulations of short time scale phenomena because of explicit time step stability limitations and equation decoupling. We developeedd an implicit algorithm with the capability to simulate physics of any length time scale because the time step is chosen by the user to match the physics of interest. This algorithm has the additional advantage that the equations are solved in a fully coupled manner. The plasma is assumed to be composed of a neutral fluid, ion fluid, and electron fluid. Each fluid has an associated temperature and can exchange energy to the other fluids by ionization and other collisional processes. The plasma is allowed to have a variable degree of ionization, from a fully ionized plasma to a completely neutral gas. The algorithm is implemented using arbitrary finite volumes so it can model realistic three-dimensional geometries.

To speed development of this effort an existing MHD code was used. *WARP3* (Washington Approximate Riemann Plasma code for 3-d domains) is a time-dependent, three-dimensional, arbitrary-geometry MHD algorithm with viscous and resistive effects. The code was extended to include thermal diffusion for the constituent temperatures (neutrals, ions, and electrons). A time-dependent ionization model was added which self-consistently calculates the ionization fraction of the fluid. Energy loss mechanisms were added for the constituent fluid components. These features were benchmarked against analytical results. The new algorithm is solved using a domain decomposition technique for parallel computation.

The implicit formulation has been developed for the resistive and viscous MHD model. The culmination of this research effort produced the Ph. D. dissertation of B. Udrea[1] and the M. S. thesis for W. Vuillemot. The algorithm has been cast using finite volumes which significantly reduces transverse flux errors. An important result of this work is the development of a 2-level nested iteration technique which accurately solves the MHD equations with typical Courant numbers of 100. The residual of the error is driven to machine accuracy for all cases investigated.

As a result of this project several professional collaborations now exist between the Department of Aeronautics and Astronautics at the University of Washington and the Air Force Research Laboratory, Lawrence Livermore National Laboratory, the University of Michigan, the University of Colorado, Stanford University, and other departments at the University of Washington. The work from this project has been presented at international conferences and published in a refereed journal.

2 Project Description

Plasmas are essential to many technologies that are important to the Air Force, some of which have dual-use potential. These applications include portable pulsed power sys-

tems, high power microwave devices, drag reduction for hypersonic vehicles, advanced plasma thrusters for space propulsion, nuclear weapons effects simulations, radiation production for counter proliferation, and fusion for power generation. Several of these applications are specifically mentioned in the *New World Vistas* Report from the USAF Scientific Advisory Board.[2] In general, plasmas fall into a density regime where they exhibit both collective (fluid) behavior and individual (particle) behavior. Many plasmas of interest can be modeled by treating the plasma like a conducting fluid and assigning macroscopic parameters that accurately describe its particle-like interactions. Magnetohydrodynamic models the plasma in this manner.

2.1 Research Objectives

The objectives of the project are to:

- Develop an implicit, conservative multi-temperature algorithm for three-dimensional non-ideal MHD simulations for time-dependent and steady state variably ionized plasmas;
- Validate the code with analytical and experimental data; and
- Apply the code to analyze plasma related topics at the Air Force Research Laboratories [the magnetic flux compression generator (MCG) experiments and the liner implosion system (WFX)[3] at Kirtland AFB, the plasma thruster work at Edwards AFB, and the hypersonic drag reduction research at Wright-Patterson AFB] and at the University of Washington [Z-Pinch experiment (ZaP)[4] and Helicity Injected Tokamak (HIT)[5]].

2.2 Technical Description

The three-dimensional, extended MHD plasma model is a set of mixed hyperbolic and parabolic equations. The Navier-Stokes equations are also of this type. This project applies some advances that have been made in implicit algorithms for the Navier-Stokes equations to the MHD equations. These implicit algorithms solve the equation set in a fully coupled manner, which generates better accuracy than the current methods used for MHD simulations.

When expressed in conservative, non-dimensional form, the MHD model is described by the following equation set.

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \mathbf{B} \\ e \end{bmatrix} + \nabla\cdot\begin{bmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v} - f_i\mathbf{B}\mathbf{B} + (p + f_i\mathbf{B}\cdot\mathbf{B}/2)\,\bar{\bar{\mathbf{I}}} \\ f_i(\mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v}) \\ (e + p + \mathbf{B}\cdot\mathbf{B}/2)\,\mathbf{v} - (\mathbf{B}\cdot\mathbf{v})\,\mathbf{B} \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\rho\mathbf{v}}_{visc} \\ \dot{\mathbf{B}}_{res} + \dot{\mathbf{B}}_{Hall} + \dot{\mathbf{B}}_{diamag} \\ P_{visc} + P_{res} + P_{cond} + P_{rad} \end{bmatrix} \tag{1}$$

The variables are density (ρ), velocity ($\mathbf{v}$), magnetic induction ($\mathbf{B}$), ionization fraction (f_i), pressure (p), and energy density (e). The energy density is

$$e = \frac{p}{\gamma - 1} + \rho\frac{\mathbf{v}\cdot\mathbf{v}}{2} + \frac{\mathbf{B}\cdot\mathbf{B}}{2} \tag{2}$$

where $\gamma = c_p/c_v$ is the ratio of the specific heats. The pressure is the total material pressure, which is the sum of the partial pressures from the neutral, ion, and electron fluids.

$$p = n_n kT_n + n_i kT_i + n_e kT_e \tag{3}$$

where k is the Boltzmann constant, n_n is the neutral number density, and T_n is the neutral temperature. The remaining variables are the ion and electron number densities and temperatures.

$$\rho = (n_n + n_i)M_i \tag{4}$$

and

$$n_e = n_i. \tag{5}$$

The number densities are determined from a time-dependent ionization model

$$\frac{dn_i}{dt} = n_e \left[< \sigma v >_{ion} n_n - < \sigma v >_{recomb} n_i \right], \tag{6}$$

where $< \sigma v >_{ion}$ is the ionization rate parameter and $< \sigma v >_{recomb}$ is the recombination rate parameter.[6] The multiple temperatures evolve independently based on the appropriate components of the energy equation and energy transfer between species.

The right hand side of eqn(1) contains the non-ideal effects. These effects include viscosity, resistivity, Hall currents, diamagnetic currents, thermal conduction, and radiation cooling. The non-ideal terms are defined by

$$\dot{\rho \mathbf{v}}_{visc} \equiv \frac{1}{ReAl} \nabla \cdot \bar{\bar{\tau}} \tag{7}$$

$$\dot{\mathbf{B}}_{res} \equiv -\frac{1}{RmAl} \nabla \times (\bar{\bar{\eta}} \cdot (\nabla \times \mathbf{B})) \tag{8}$$

$$\dot{\mathbf{B}}_{Hall} \equiv -\frac{\omega_{ce}\tau_e}{RmAl} \nabla \times \left[\frac{(\nabla \times \mathbf{B}) \times \mathbf{B}}{\rho} \right] \tag{9}$$

$$\dot{\mathbf{B}}_{diamag} \equiv \frac{\omega_{ce}\tau_e}{RmAl} \nabla \left(\frac{1}{\rho} \right) \times \nabla p_e \tag{10}$$

$$P_{visc} \equiv \frac{1}{ReAl} \nabla \cdot \mathbf{v} \cdot \bar{\bar{\tau}} \tag{11}$$

$$P_{res} \equiv -\frac{1}{RmAl} \nabla \cdot \bar{\bar{\eta}} \cdot (\nabla \times \mathbf{B}) \times \mathbf{B} \tag{12}$$

$$P_{cond} \equiv \frac{M_i}{2PeAl} \nabla \cdot \bar{\bar{k}} \cdot (\nabla T_n + \nabla T_i + \nabla T_e) = P_{cond_n} + P_{cond_i} + P_{cond_e} \tag{13}$$

$$P_{rad} \equiv -C_{rad} Z_{eff} (\rho f_i)^2 T_e^{1/2} \tag{14}$$

M_i is the ion mass, $\omega_{ce}\tau_e$ is the Hall parameter, C_{rad} is the Bremsstrahlung radiation constant, and Z_{eff} is the effective ionization level due to plasma impurities.

The non-dimensional tensors are the stress tensor ($\bar{\bar{\tau}}$), the electrical resistivity ($\bar{\bar{\eta}}$), and the thermal conductivity ($\bar{\bar{k}}$), and $\bar{\bar{I}}$ is the identity matrix. The non-dimensional numbers are defined as follows:

$$\begin{array}{lll} \text{Alfvén Number :} & Al & \equiv V_A/V \\ \text{Reynolds Number :} & Re & \equiv LV/\nu \\ \text{Magnetic Reynolds Number :} & Rm & \equiv \mu_o LV/\eta \\ \text{Péclet Number :} & Pe & \equiv LV/\kappa \end{array} \tag{15}$$

The characteristic variables are length (L), velocity (V), Alfvén speed ($V_A = B/\sqrt{\mu_o \rho}$), kinematic viscosity ($\nu$), electrical resistivity (η), and thermal diffusivity ($\kappa = k/\rho c_p$). μ_o is the permeability of free space ($4\pi \times 10^{-7}$).

For convenience, the MHD equation set [eqn(1)] is rewritten in the following compact form

$$\frac{\partial Q}{\partial t} + \nabla \cdot \bar{\bar{T}}_h = \dot{Q}_{Non-ideal} \tag{16}$$

where Q is the vector of conservative variables, $\bar{\bar{T}}_h$ is the tensor of hyperbolic fluxes, and $\dot{Q}_{Non-ideal}$ contains the non-ideal (mostly parabolic) terms. The forms of these vectors and tensors can be seen from the previous equations. The hyperbolic fluxes are associated with wave-like motion, and the parabolic fluxes are associated with diffusion-like motion.

2.3 Multiple Temperature Evolution

The temperatures of the multiple species (neutrals, ions, electrons) evolve independently based on the appropriate components of the energy equation and energy transfer between species. The temperatures are consistent with energy conservation.

The temperature rise in each specie depends on the heating mechanism and the density fraction of the specie. We define the density fractions as

$$f_i = \frac{n_i}{n_n + n_i}, \tag{17}$$

$$f_n = \frac{n_n}{n_n + n_i} = 1 - f_i, \tag{18}$$

and

$$f_e = f_i. \tag{19}$$

For multiply charged species, the last definition would be modified to $f_e = Z f_i$. Viscous drag heats the neutral gas and the ion fluid, but does not affect the electrons. Therefore, the energy rise due to viscous effects is attributed to the neutral and ion temperatures.

$$\left.\frac{\partial p_n}{\partial t}\right|_{visc} = (\gamma - 1) f_n P_{visc} \tag{20}$$

$$\left.\frac{\partial p_i}{\partial t}\right|_{visc} = (\gamma - 1) f_i P_{visc} \tag{21}$$

Resistivity directly heats only the electrons.

$$\left.\frac{\partial p_e}{\partial t}\right|_{res} = (\gamma - 1) P_{res} \tag{22}$$

Radiation is emitted by the electrons as they cool.

$$\left.\frac{\partial p_e}{\partial t}\right|_{rad} = (\gamma - 1) P_{rad} \tag{23}$$

Each species has its own thermal conduction component as is evident from eqn(13).

$$\left.\frac{\partial p_n}{\partial t}\right|_{cond} = (\gamma - 1) P_{cond_n} \tag{24}$$

$$\left.\frac{\partial p_i}{\partial t}\right|_{cond} = (\gamma - 1) P_{cond_i} \tag{25}$$

$$\left.\frac{\partial p_e}{\partial t}\right|_{cond} = (\gamma - 1) P_{cond_e} \tag{26}$$

The remaining total material pressure rise is due to adiabatic compression which affects all species in proportion to their densities.

Energy may also transfer between species due to collisions. However, total energy is conserved and the total material pressure is not affected. The interspecies energy transfer is modeled to preserve the total material pressure.

$$p = p_n + p_i + p_e = p'_n + p'_i + p'_e \tag{27}$$

2.4 Conservative Algorithm

Because of the natural differences between hyperbolic and parabolic equations, the methods for solving them are very different. Since the MHD equations are of mixed type the hyperbolic and parabolic terms must be handled differently. The hyperbolic fluxes are differenced by applying an implicit, approximate Riemann algorithm that properly accounts for their wave-like behavior. The parabolic terms are discretized by applying explicit central differencing. The remaining non-ideal terms which correspond to the Hall effect are solved using a semi-implicit method.[7]

The design of the algorithm is driven by the conservative numerical techniques that must be used for the hyperbolic terms. Therefore, we begin by considering the ideal MHD equations, which are obtained from eqn(16) by setting all the non-ideal terms ($\dot{Q}_{Non-ideal}$) to zero. Note that ideal MHD refers to an ideal plasma — one that is inviscid and non-resistive and neglects thermal conduction and finite Larmor radius (FLR) effects.

The ideal MHD equations are

$$\frac{\partial Q}{\partial t} + \nabla \cdot \bar{\bar{T}}_h = \frac{\partial Q}{\partial t} + A \nabla \cdot Q = 0, \tag{28}$$

where A is the Jacobian of the hyperbolic flux tensor.

$$A = \frac{\partial \bar{\bar{T}}_h}{\partial Q} \tag{29}$$

Here, Q is the vector of conserved variables.

$$Q = (\rho, \rho v_x, \rho v_y, \rho v_z, B_x, B_y, B_z, e)^T. \tag{30}$$

This is a set of hyperbolic equations and thus A_x has a complete set of real eigenvalues given by

$$\lambda = (v_x, 0, v_x \pm V_{fast}, v_x \pm V_{slow}, v_x \pm V_{Ax})^T, \tag{31}$$

where V_{fast} and V_{slow} are the fast and slow magnetosonic speeds, and V_{Ax} is the Alfven speed based on the x component of the magnetic field. These can be expressed as

$$V_{fast}^2 = \frac{1}{2}\left[c_s^2 + V_A^2 + \sqrt{(c_s^2 + V_A^2)^2 - 4c_s^2 V_{Ax}^2}\right], \tag{32}$$

$$V_{slow}^2 = \frac{1}{2}\left[c_s^2 + V_A^2 - \sqrt{(c_s^2 + V_A^2)^2 - 4c_s^2 V_{Ax}^2}\right], \tag{33}$$

$$V_{Ax}^2 = \frac{B_x^2}{\mu_o \rho}. \tag{34}$$

Here, c_s is the ion sound speed, which for a perfect gas is

$$c_s^2 = \frac{\gamma p}{\rho}. \tag{35}$$

We make special note of the zero eigenvalue, λ_2 in this case. The zero eigenvalue only appears in multiple dimensions and is caused by the perpendicular nature of the $\mathbf{j} \times \mathbf{B}$ force. Powell *et al.*,[8] recently solved this zero eigenvalue problem by introducing a source term that is proportional the divergence of the magnetic field. The eigenvalue becomes finite, $\lambda_2 = v_x$ in this case. We have implemented this modification and it is discussed in a later section.

For hyperbolic equations information propagates along characteristics which travel at wave speeds given by the eigenvalues. Most modern numerical techniques for solving hyperbolic equations are based upon the idea of splitting the fluxes into components due to left- and right-running waves. Then each part of the flux can be differenced in an upwind manner, which greatly reduces numerical oscillations and stabilizes the solutions.

It is well known that if a hyperbolic equation is solved with an explicit scheme, then the allowable time step to maintain numerical stability is given by the CFL (Courant-Friedrichs-Lewy) condition, which in the case of the 1D MHD equations is

$$\Delta t < \frac{\Delta x}{|v_x + V_{fast}|}. \tag{36}$$

For the high magnetic fields and low densities common in many plasma experiments, the fast magnetosonic speed is quite high, and thus the time step is prohibitively small. We

are often interested in only modeling the physics that occurs slower than Alfvén time scales. For example, it can be shown that resistive tearing modes, which are important in studying fusion plasmas, evolve on a time scale that is given by[9]

$$\tau_{tearing} \propto \tau_A^{2/5} \tau_\eta^{3/5} = (Lu)^{3/5} \tau_A. \tag{37}$$

τ_A is the Alfvén time, τ_η is the resistive diffusion time, and Lu is the Lundquist number, which is given by

$$Lu = \frac{\tau_\eta}{\tau_A} = RmAl. \tag{38}$$

If Lu is 10^6, which is typical for laboratory plasmas in fusion applications, the resistive tearing time is approximately 4000 times larger than the Alfvén time. By treating the hyperbolic fluxes implicitly in time, the stability restriction on the time step is removed, and the solution can be advanced at the larger resistive tearing time step. This is our motivation for proposing an implicit scheme.

2.5 Implicit Formulation

For clarity, the algorithm for the two-dimensional ideal MHD equations is presented. The extension to three dimensions is straight forward. The two-dimensional equation is

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0. \tag{39}$$

Eqn(39) was discretized using first order Euler time differencing to get

$$\frac{\left(Q_{ij}^{n+1} - Q_{ij}^{n}\right)}{\Delta t} = -R_{ij}\left(Q^{n+1}\right) = -R_{ij}^{n+1} \tag{40}$$

where R is

$$R_{ij} = F_{i+1/2,j} - F_{i-1/2,j} + G_{i,j+1/2} - G_{i,j-1/2}. \tag{41}$$

Note that in this equation and all that follow the grid metric terms (cell areas and volumes) were omitted for clarity. Linearizing R as follows:

$$R_{ij}^{n+1} \approx R_{ij}^{n} + \left(\frac{\partial R}{\partial Q}\right)_{ij}^{n} \left(Q_{ij}^{n+1} - Q_{ij}^{n}\right) \tag{42}$$

where $\partial R/\partial Q$ has been defined as the differenced flux Jacobians.

$$\frac{\partial R}{\partial Q} == \left.\frac{\partial F}{\partial Q}\right|_{i+1/2,j} - \left.\frac{\partial F}{\partial Q}\right|_{i-1/2,j} + \ldots \tag{43}$$

where $\partial F/\partial Q$ is the flux Jacobians of the x flux. The flux Jacobians can be calculated analytically or numerically. Analytical calculations based on the assumption that the solution values do not change rapidly were used previously[10] and produced adequate results. The current project investigated numerical calculation of the flux Jacobians and an analytical method without the previous assumption. Two methods for numerical

calculation were investigated. First, a limit formulation similar to the definition of a differential was used.

$$\frac{\partial F}{\partial Q} = \frac{F(Q+\epsilon) - F(Q)}{\epsilon} + O(\epsilon) \tag{44}$$

for small ϵ which gave first order accuracy in ϵ. The flux Jacobians were also calculated using complex numbers. The flux Jacobians were expanded about Q in a Taylor series.

$$F(Q+ih) = F(Q) + ih\frac{\partial F}{\partial Q} - \frac{h^2}{2}\frac{\partial^2 F}{\partial Q^2} - i\frac{h^3}{6}\frac{\partial^3 F}{\partial Q^3} + ... \tag{45}$$

The expression was rearranged to solve for the flux Jacobian.

$$\frac{\partial F}{\partial Q} = \Im\left[\frac{F(Q+ih)}{h}\right] + O(h^2) \tag{46}$$

which gave second order accuracy in h. Additionally, the complex formulation required only a single evaluation of the flux Jacobian (though using complex math) compared to two evaluations for the limit formulation.

Substituting the expression for R_{ij}^{n+1} back into eqn(40) and rearranging, gave

$$\left[\frac{\bar{\bar{\mathrm{I}}}}{\Delta t} + \left(\frac{\partial R}{\partial Q}\right)_{ij}^{n}\right] \Delta Q_{ij}^{n} = -R_{ij}^{n}. \tag{47}$$

Here ΔQ has been defined as

$$\Delta Q^n \equiv Q_{ij}^{n+1} - Q_{ij}^{n}. \tag{48}$$

The left hand side of the eqn(47) is an implicit operator operating on ΔQ. It is a large banded block matrix. In three dimensions, it is an $(I_{max} \times J_{max} \times K_{max})$ by $(I_{max} \times J_{max} \times K_{max})$ matrix, where I_{max} is the number of cells in the x direction, etc. It is quite costly to invert a matrix of this size directly. For this project a more efficient iterative method was chosen. When solved iteratively , eqn(47) can lose time accuracy. To recover time accuracy the time derivative of Q was added as a source term to the right hand side of the equation. The modified equation became

$$\left(\frac{\partial Q}{\partial \tau}\right)^{n+1} = -R_{ij}^{n+1} - S_{ij}^{n+1} \tag{49}$$

where

$$S_{ij}^{n+1} = \frac{1}{2\Delta t}\left(3Q_{ij}^{n+1} - 4Q_{ij}^{n} + Q_{ij}^{n-1}\right) \approx \frac{\partial Q}{\partial t}. \tag{50}$$

The τ in eqn(49) is an iteration variable,called pseudo time. At each physical time step, eqn(49) is solved iteratively until the left hand side vanishes. When the solution converges, our original equation

$$\frac{\partial Q}{\partial t} = -R \tag{51}$$

is solved. This technique is known as dual time-stepping.[11] Note that in eqn(50) a more accurate time derivative can be used at the expense of the additional memory needed

to store the Q vectors from previous time steps. A concern for higher order numerical methods is the property of total variation diminishing (TVD). The TVD property prevents some types of nonlinear numerical instabilities. The dual time-stepping method has not been shown to be TVD for the MHD model.

One advantage of the strategy outlined above is that the implicit operator and the right hand side in eqn(47) are decoupled. The structure of the matrix no longer depends on the details of the discretization of the right hand side fluxes. The iteration equation [eqn(47)]

$$\left[\frac{3\bar{\bar{\mathrm{I}}}}{3\Delta\tau + 2\Delta t} + \left(\frac{\partial R}{\partial Q}\right)^{m}_{ij}\right] \Delta Q^{m}_{ij} = -\frac{1}{2\Delta t}\left(3Q^{m}_{ij} - 4Q^{n}_{ij} + Q^{n-1}_{ij}\right) - R^{m}_{ij} \tag{52}$$

has the standard form

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{53}$$

Previously the LU-SGS method (lower-upper symmetric Gauss-Seidel)[12] was used to invert the implicit operator$\mathbf{A}$.[13, 10] The LU-SGS method required a modification of the implicit operator through an approximate factorization procedure which reduced the accuracy of the operator and led to poor convergence.

The current project used a symmetric Gauss-Seidel method which does not rely on approximate factorization. The SGS method was used to iteratively invert the implicit operator and the approximate Riemann solver that is used to form the right hand side fluxes.

The implicit operator matrix was decomposed into lower, upper, and diagonal matrices and written as

$$\mathbf{A} = \mathbf{L} + \mathbf{U} + \mathbf{D} \tag{54}$$

Each iteration of the symmetric Gauss-Seidel method performs two sweeps of the domain — a forward sweep followed by a backward sweep.

$$(\mathbf{L} + \mathbf{D})\mathbf{x}^{(2l-1)} + \mathbf{U}\mathbf{x}^{(2l-2)} = \mathbf{b} \tag{55}$$

$$(\mathbf{U} + \mathbf{D})\mathbf{x}^{(2l)} + \mathbf{L}\mathbf{x}^{(2l-1)} = \mathbf{b} \tag{56}$$

where $l = 1, 2, 3, \ldots$ is the iteration index.

For reasonably large values of Re and Rm (easily in the range of interest for most applications), the parabolic terms can be differenced explicitly without constraining the allowable time step. The right hand side of eqn(47) was modified by adding the central differenced parabolic terms.

2.6 Approximate Riemann Solver

The fluxes on the right hand side of eqn(47) were discretized using a Roe-type approximate Riemann solver.[14] In this method the overall solution was built upon the solutions to the Riemann problem defined by the discontinuous jump in the solution between each pair of cells. The numerical flux for a first-order accurate (in space) Roe-type solver was written in symmetric form as

$$F_{i+1/2} = \frac{1}{2}\left(F_{i+1} + F_i\right) - \frac{1}{2}\sum_k l_k \left(Q_{i+1} - Q_i\right) |\lambda_k| \, r_k \tag{57}$$

where r_k is the kth right eigenvector, λ_k is the absolute value of the kth eigenvalue, and l_k is the kth left eigenvector. The values at the cell interface $(i + 1/2)$ were obtained by either a simple average or, more accurately, a Roe-average of the neighboring cells. Determining a Roe-average on an arbitrary computational grid involved transforming the vector quantities to a coordinate system that is orthogonal to the local cell interface. Then the flux calculated as above is normal to the cell interface which is the desired orientation for applying the divergence theorem in a finite volume method.

These first order accurate upwind fluxes are used in the vicinity of sharp discontinuities in order to suppress oscillations in the solution. Globally second order accurate solutions were achieved by using a flux limiter that modifies the first order flux so that it uses second order central differencing in smooth portions of the flow. The minmod limiter was used.[15]

Once the eigenvalues and eigenvectors were obtained and properly normalized to avoid singularities, it was relatively straight-forward to apply this scheme to the one-dimensional ideal MHD equations.[16, 17] Unlike for the Euler equations, the extension to more than one dimension was non-trivial. In more than one dimension, the Q vector must include B_x in addition to the other magnetic field components. (For the one-dimensional case B_x is constant by virtue of $\nabla \cdot \mathbf{B} = 0$). Since the $\mathbf{j} \times \mathbf{B}$ force acts perpendicularly to the directions of $\mathbf{j}$ and $\mathbf{B}$, the F flux vector has a zero term corresponding to B_x. Thus, the Jacobian matrix of F is singular and has a zero eigenvalue. A complete set of physically meaningful eigenvectors no longer exists. Physically, one would expect information to travel either at the fluid velocity or at the fluid velocity plus or minus the wave speeds. Simply dropping B_x from the equation set is not a viable option, because B_x needs to change in order to maintain $\nabla \cdot \mathbf{B} = 0$. Powell *et al.*,[8] solved this problem by modifying the Jacobian in such a way as to change the zero eigenvalue to v_x (keeping the others unchanged), and then adding in a source term that exactly canceled out the terms introduced by the modified Jacobian.

The source term is

$$S_{div} = -\begin{bmatrix} \rho \\ \mathbf{B} \\ \mathbf{v} \\ \mathbf{v} \cdot \mathbf{B} \end{bmatrix} \nabla \cdot \mathbf{B} \tag{58}$$

It is proportional to the divergence of $\mathbf{B}$ and thus very small. The effect of this modification is to sweep the field divergence out of the domain with the plasma flow (for example, $\lambda_k = v_x$). However, for closed boundaries or stagnation points the divergence increases. A divergence cleaner has been implemented based on the Hodge projection technique.

2.7 Finite Volume Grid and Parallel Implementation

Since the algorithm being developed will be used for real devices, it must be capable of modeling arbitrary, three-dimensional geometries. Therefore, multi-block, finite volume grids were used. A typical cell is shown in Figure 1. The computational domain is divided into blocks which are then spanned by body-fitting finite volume cells. See Figure 2 for a possible grid.

As discussed in the previous section, the formulation of the approximate Riemann solver which we have developed generates hyperbolic fluxes oriented normal to the grid cell interfaces. Application of the divergence theorem is then a simple operation. The

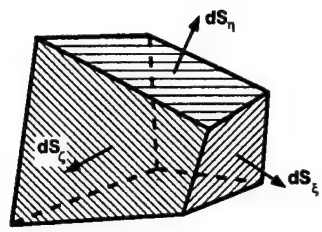

Figure 1: Schematic of the arbitrary shaped three-dimensional finite volume cell used by the algorithm.

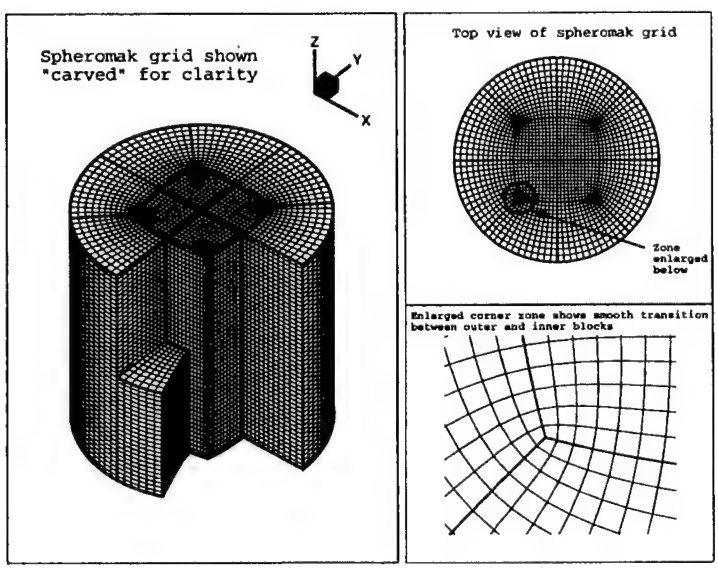

Figure 2: Schematic of the arbitrary shaped three-dimensional finite volume cell used by the algorithm.

parabolic fluxes are also calculated to be normal to the grid cell interfaces. To accomplish this orientation, a set of nested control volumes were used and the appropriate vector operations within these volumes were applied.

The block structure of the grid provided a natural domain decomposition for the parallel implementation. The integral form of a general conservation law was expressed as

$$\frac{\partial}{\partial t}\int_{\Omega} dV\, Q + \oint_{\Sigma} dS \cdot F(Q) = \int_{\Omega} dV\, S(Q), \tag{59}$$

where Ω is the domain and Σ is the boundary of Ω. Q is the vector of conserved variables, $F(Q)$ is the flux of the conserved variables, and $S(Q)$ is the vector of source terms. By splitting the domain Ω into p subdomains such that

$$\Omega = \bigcup_{i=1}^{p} \Omega_i, \tag{60}$$

eqn(59) was replaced with a set of p conservation equations applied on the subdomains Ω_i.

$$\frac{\partial}{\partial t}\int_{\Omega_i} dV\, Q + \oint_{\Sigma_i} dS \cdot F(Q) = \int_{\Omega_i} dV\, S(Q), \quad i = 1, 2, ..., p \tag{61}$$

Each of these discretized equations is solved by a single processor using the boundary values copied from neighboring subdomains.

To ensure a portable code a message passing system commonly available on parallel supercomputers and on workstation clusters was used. This system was the Message Passing Interface (MPI)[18], which was adopted as a standard in May 1994 by industry and academia. Hardware and software vendors' implementation of MPI provides parallel program developers with a consistent set of subroutines callable from FORTRAN90 and C. In this project the basic point-to-point communications subroutines and global communications subroutines were used. The point-to-point communication subroutines were used for the domain decomposition and boundary exchange while the global communication subroutines were used for convergence checking. All message passing systems (PVM, MPL) support point-to-point and global communications subroutines so that by using only the basic set portability to systems not supporting MPI was simplified.

3 Project Implementation and Results

3.1 Finite Volume Improvements

To improve the codes ability to handle highly distorted grids, finite volume grids were implemented. The finite volume implementation greatly reduced the anomalous momentum leakage into orthogonal directions when the grid was distorted.

Figure 3 shows a shock tube test problem. The simulation was performed in three-dimensions, but should remain one-dimensional. The figure shows the three gas dynamic waves in the density plot. The transverse velocity should be zero. A finite amount of momentum leakage was generated by the grid metrics in the finite difference generalized coordinate formulation.

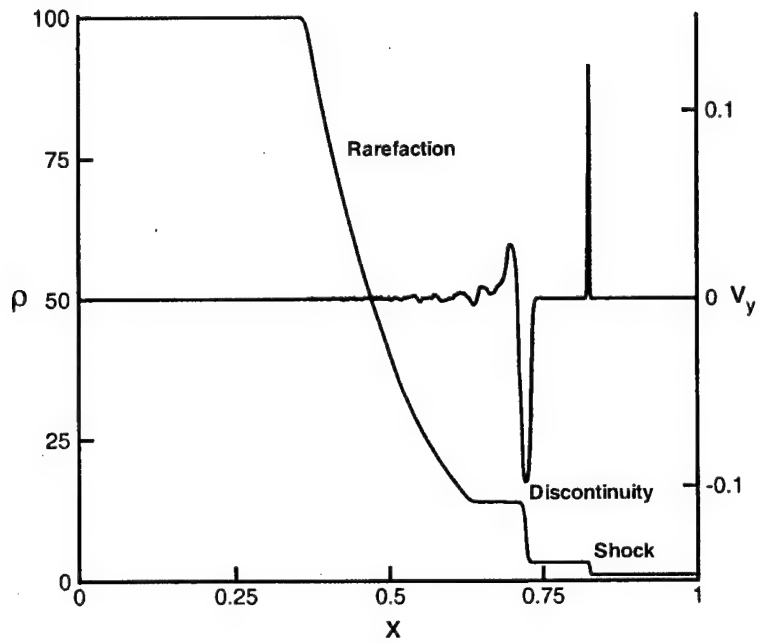

Figure 3: A three-dimensional simulation of a one-dimensional shock tube showing the density and transverse velocity. The three gas dynamic waves can be seen in the density plot. The transverse velocity should be zero.

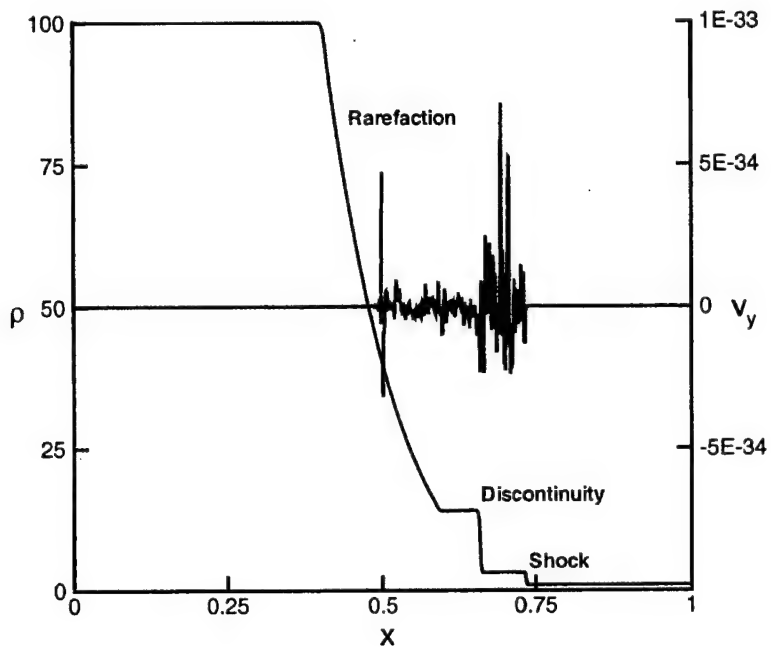

Figure 4: A three-dimensional simulation of a one-dimensional shock tube showing the density and transverse velocity. The three gas dynamic waves can be seen in the density plot. The transverse velocity should be zero.

The reduction of the anomalous momentum leakage into orthogonal directions can be seen in Figure 4. The simulation was identical to that shown in Figure 3 except a finite volume implementation was used instead of the generalized coordinate formulation. Figure 4 shows the same three gas dynamic waves in the density plot. With the finite volume implementation, the transverse velocity is zero to within machine accuracy.

3.2 Implicit Formulation and Numerical Flux Jacobian Calculations

The LU-SGS method (lower-upper symmetric Gauss-Seidel)[12] was used previously to invert the implicit operator of eqn(47).[13, 10] The LU-SGS method required a modification of the implicit operator through an approximate factorization procedure which reduced the accuracy of the operator and led to poor convergence. The convergence history is shown in Figure 5 The explanation for the poor convergence was inaccurate approximation of the implicit operator. The inaccuracy developed from the combination of the approximate factorization and the approximate analytical calculation of the flux Jacobians.

The standard formulation of eqn(52) allowed the use of standard iterative matrix

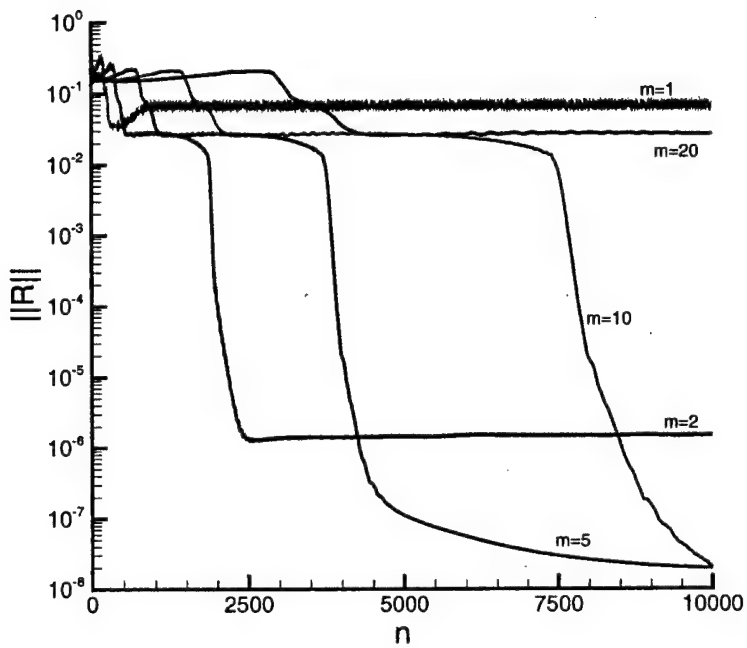

Figure 5: Convergence history using the LU-SGS method to invert the implicit operator. n is the number of physical time iterations, and m is the number of LU-SGS pseudo time subiterations.

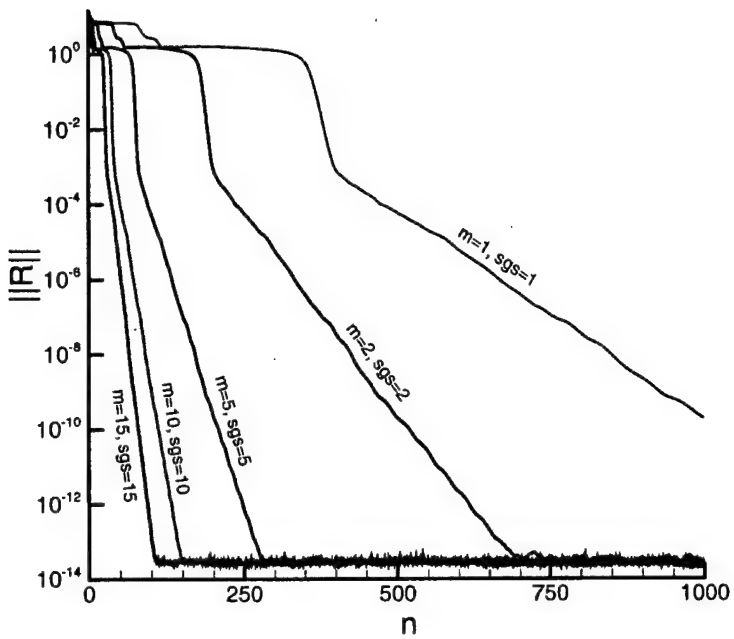

Figure 6: Convergence history using the SGS method to invert the implicit operator. n is the number of physical time iterations, m is the number of pseudo time subiterations, and sgs is the number of iterations of the SGS method.

inversion methods. In this project the symmetric Gauss-Seidel method was used. The convergence history is shown in Figure 6. n is the number of physical time iterations, m is the number of pseudo time subiterations, and sgs is the number of iterations of the SGS method. Unlike the LU-SGS method, there is no coupling between the SGS iterations and the pseudo time iterations. The values of the implicit operator $\mathbf{A}$ and the inhomogeneity vector $\mathbf{b}$ are updated between pseudo time iterations, but not between SGS iterations.

The formulation of eqn(52) into a standard form required numerical calculation of the flux Jacobians. The limit calculation given by eqn(44) was simple to implement and gave accurate values of the flux Jacobian. However, it was sensitive on the value of ϵ when ϵ was increased beyond 1×10^{-12}. Using the complex formulation given by eqn(46) provided accurate flux Jacobian calculations with much less sensitivity on h. Typical values of h were 1×10^{-5}.

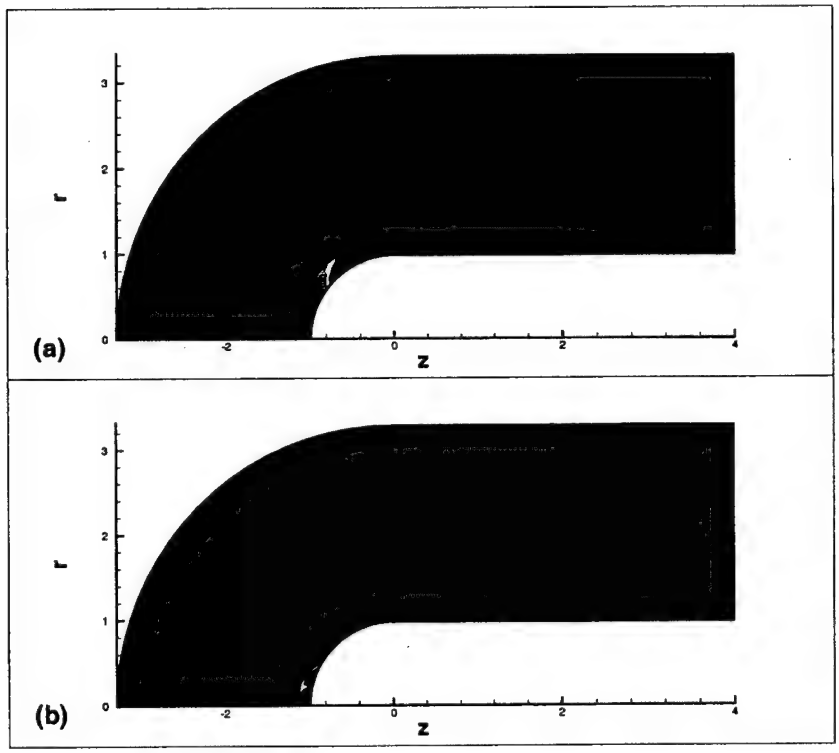

Figure 7: Mach 6 flow impinging on a hemispherical body. The upper plot (a) shows the density contours. The lower plot (b) shows the ionization fraction. Notice the increased ionization at the stagnation point.

3.3 Time Dependent Ionization and Multiple Temperature Effects

A time-dependent ionization model was added to self-consistently calculate the ionization fraction of the fluid. The model is described by eqn(6). The ionization rate parameter $< \sigma v >_{ion}$ and the recombination rate $< \sigma v >_{recomb}$ were calculated using empirical formulations given in Ref.[6].

The time-dependent ionization model allowed determination of the ionization fraction. For uniform flow properties the time dependence is exponential. The model was benchmarked against analytical formulations for its time-dependence. The steady-state solution was benchmarked against the Saha equation.

A more interesting test was constructed to have a Mach 6 flow impinge on a hemispherical body. The flow was initially unionized, and ionized upon transition through the shock wave. The results are shown in Figure 7.

Multi-temperature effects have been added to the code. The code was extended to include thermal diffusion for the constituent temperatures (neutrals, ions, and electrons). Energy loss mechanisms were added for the constituent fluid components. The constituent temperatures can evolve independently by diffusion as described in eqn(13). The heat conduction equation was used as a benchmark to validate the incorporation

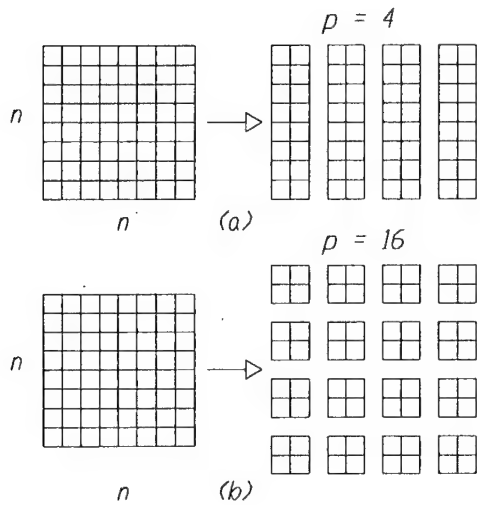

Figure 8: (a) Strip decomposition and (b) patch decomposition of a 2-D domain.

of the multi-temperature thermal diffusion model.

A dominate energy loss mechanism for high electron temperature plasmas is radiation. The radiation loss term due to Bremsstrahlung has been included. The radiation was validated against analytical results.

3.4 Parallel Computer Performance

The algorithm was parallelized using the domain decomposition technique (DDT). The integral form of a general conservation law was given by eqn(59). Domain decomposition implementation requires boundary (or ghost) cells to overlap with neighboring domains or blocks. The domain decomposition is illustrated for two dimensions in Figure 8.

The ghost cells are used as boundary conditions to the real cells in the block. A consequence of domain decomposition is the more blocks that are used the more ghost cells that are necessary. The ghost cell data lag the current computation by a single iteration. Therefore, an increase on ghost cells generated a slower convergence rate. Figure 9 shows the slightly slower convergence rate. The convergence history for 4 and 8 processors overlay.

The parallel performance for the code is shown in Figures 10 and 11 for the code operating in explicit and implicit mode. The grid was the three-dimensional grid shown in Figure 2 and was parallelized using domain decomposition. The grid was scaled with the number of processors, so the grid size per processor was constant. As the number of processors was doubled, the number of grid cells was also doubled. The ideal speedup was unity. Note that the speedup presented is "engineering" speedup. The value includes not only the inefficiencies associated with communication between the processors but also those associated with more iterations required to converge the solution. The "engineering" speedup is, therefore, the total parallel efficiency to obtain the same quality of solution on a parallel computer.

The high parallel efficiency was obtained by overlapping communication with com-

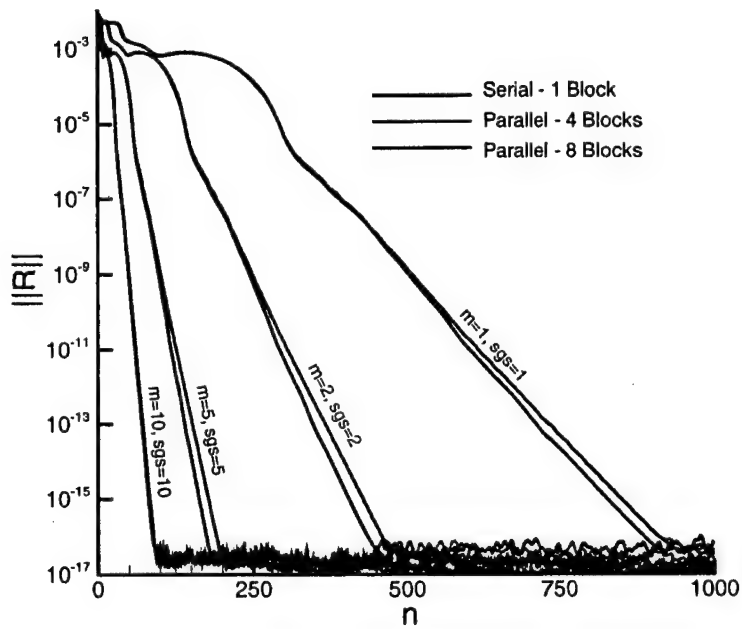

Figure 9: Convergence history using the SGS method to invert the implicit operator on a parallel computer. The results from a serial computer are plotted for comparison. n is the number of physical time iterations, m is the number of pseudo time subiterations, and sgs is the number of iterations of the SGS method.

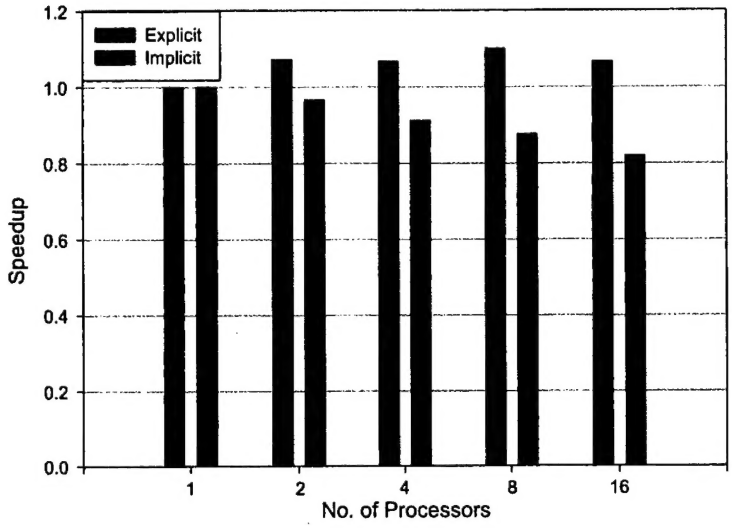

Figure 10: Parallel speedup for a three-dimensional grid using domain decomposition on a cluster of DEC Alpha workstations. The grid is scaled with the number of processors, so the grid size per processor is constant. The ideal speedup is unity.

putation. The boundary information was exchanged while the core cells were computed and before the boundary cells are computed. The super-linear speedup for the explicit operating mode was generated by slow operation on a single processor. This effect has been reported Michl *et al.*[19] Figure 10 contains the performance results from a parallel workstation cluster of 16 DEC Alpha workstations. Figure 11 contains the performance results from the IBM SP2 at the Maui High Performance Computing Center. The results on both computing platforms were similar.

4 Professional Interactions

4.1 Project Personnel

The personnel who have been directly involved in this project are listed below.

Name	Position
Uri Shumlak	Assistant Professor
D. Scott Eberhardt	Associate Professor
Thomas R. Jarboe	Professor
Chris Aberle	Graduate Student
John Loverich	Graduate Student
Ward Vuillemot	Graduate Student
Graham Schelle	Undergraduate Student

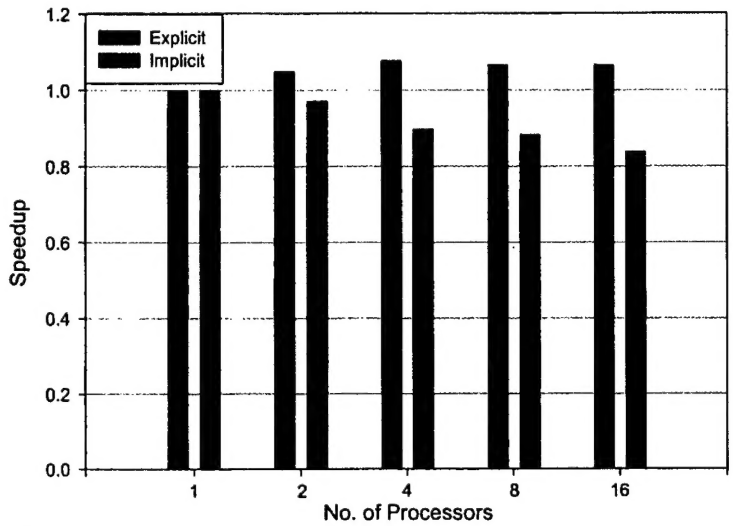

Figure 11: Parallel speedup for a three-dimensional grid using domain decomposition on IBM SP2 parallel supercomputer. The grid is scaled with the number of processors, so the grid size per processor is constant. The ideal speedup is unity.

4.2 Collaborations

4.2.1 Air Force Research Laboratory

Dr. Robert Peterkin, Jr. of the Electromagnetic Sources Division of the Air Force Research Laboratory at Kirtland AFB on three-dimensional multigrid algorithms for MACH3, development of a parallel PIC (particle in cell) code for microwave simulations, and stabilization of the Rayleigh-Taylor instability in solid liner implosions by introducing a sheared axial flow by designing conical liners. Knowledge developed in the area of relaxation schemes was implemented into the ICEPIC code to make a 3-D Poisson solver. The solver was needed to determine electric field concentration on a high power microwave source. The collaboration occured in person during August. Several phone and email discussions took place throughout the year.

4.2.2 Sandia National Laboratories

Dr. Norm Roderick of the Pulsed Power Sciences Center at Sandia National Laboratories on the uses of sheared axial flows to stabilize z-pinch implosions. This is an ongoing collaboration that resulted in the publication listed in the following section.

4.2.3 University of Washington

Prof. Scott Eberhardt of the Aeronautics and Astronautics Department and Prof. Randy LeVeque of the Applied Math Department on approximate Riemann solvers and their applications to multidimensional problems. We have regular discussions on a weekly basis.

Prof. Tom Jarboe of the Aeronautics and Astronautics Department on the higher mode stability of spheromaks and on the effect of realistic three-dimensional geometries on spheromak stability. This is an ongoing collaboration that resulted in the publications listed in the following section.

4.3 Publications

A journal article describing our algorithm has been published in the *Journal of Computational Physics.* The title is "An Implicit Scheme for Nonideal Magnetohydrodynamics" by O. S. Jones, U. Shumlak, and D. S. Eberhardt.[13] The citation is Journal of Computational Physics **130,** 231 (1997). Another journal article which describes the use of sheared flows to stabilize the Rayleigh-Taylor instability has been published in *Physics of Plasmas.* This is work that was performed with collaboration at the Air Force Research Laboratory. The title is "Mitigation of the Rayleigh-Taylor Instability by Sheared Axial Flows" by U. Shumlak and N. F. Roderick.[20] The citation is Physics of Plasmas **5,** 2384 (1998).

Two papers describing the higher mode stability in spheromak plasmas and the effect of realistic three-dimensional geometries on spheromak stability have also been published.[21, 22] The citations are Physics of Plasmas **6,** 4382 (1999) and Physics of Plasmas **7,** 2959 (2000).

Two papers describing this project will be presented at the upcoming AIAA Computational Fluid Dynamics conference. One paper titled "An Approximate Riemann Solver for MHD Computations on Parallel Architectures" will present an overview of the project, and the other paper will present recent work on the analytical flux Jacobian calculation.

5 Conclusions

The successful development of the three-dimensional advanced implicit algorithm and the implementation of time-dependent ionization and multiple temperature effects show that this project is reaching its objectives. The research related to this project has been published in refereed journals and presented at international conferences. Valuable collaborations have been formed with the Air Force Research Laboratory, Sandia National Laboratory, and other universities.

The continuing development of this project will include investigating the TVD properties and the analytical flux Jacobian methods. An important result of this work is the determining of the difficulty in stabilizing the Hall effect. A two fluid plasma model may be essential to properly treat Hall effect plasma physics.[23]

References

[1] B. Udrea, Ph. D. Thesis, University of Washington, Seattle, Washington (1999).

[2] G. H. McCall, J. A. Corder and the USAF Scientific Advisory Board, *New World Vistas, Air and Space Power for the 21st Century,* United States Air Force Document, Summary Volume (December 1995).

[3] F. M. Lehr, A. Alaniz, J. D. Beason, L. C. Carswell, J. H. Degnan, J. F. Crawford, S. E. Englert, T. J. Englert, J. M. Gahl, J. H. Holmes, T. W. Hussey, G. F. Kiuttu,

B. W. Mullins, R. E. Peterkin, Jr., N. F. Roderick, P. J. Turchi, and J. D. Graham, J. Appl. Phys. **75**, 3769 (1994).

[4] U. Shumlak and C.W. Hartman, Phys. Rev. Lett. **75**, 3285 (1995).

[5] B. A. Nelson, T. R. Jarboe, D. J. Orvis, L. McCullough, J. Xie, C. Zhang, and L. Zhou, Phys. Rev. Lett. **72**, 3666 (1994).

[6] Ya. B. Zel'dovich and Yu. P. Raizer, *Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena,* Academic Press, New York (1966).

[7] D. Harned and Z. Mikic, J. Comput. Phys. **83**, 1 (1989).

[8] K. G. Powell, B. van Leer, and P. L. Roe, *Private Communication,* 1994.

[9] H. P. Furth, J. Kileen, and M. N. Rosenbluth, Phys. Fluids **6**, 479 (1963).

[10] U. Shumlak, "Development of an Advanced Implicit Algorithm for MHD Computations on Parallel Supercomputers," Final Technical Report to AFOSR, Feb 1999.

[11] H. Ok and D. S. Eberhardt, "Solution of Unsteady Incompressible Navier-Stokes Equations Using an LU Decomposition Scheme," AIAA-91-1611 (1991).

[12] S. Yoon and A. Jameson, AIAA J. **26**, 1025 (1988).

[13] O. S. Jones, U. Shumlak, and D. S. Eberhardt, J. of Comp. Physics **130,** 231 (1997).

[14] P. L. Roe, J. Comp. Phys. **43**, 357 (1981).

[15] R. J. LeVeque, *Numerical Methods for Conservation Laws,* Birkhauser Verlag, Boston (1992).

[16] M. Brio and C. C. Wu, J. Comp. Phys. **75**, 400 (1988).

[17] A. L. Zachery and P. Colella, J. Comp. Phys. **99**, 341 (1992).

[18] Message Passing Interface Forum *MPI: A Message Passing Interface Standard,* May 5, 1994.

[19] T. Michl, S. Wagner, M. Lenke, A. Bode, *Dataparallel Implicit Navier-Stokes Solver on Different Multiprocessors* Parallel CFD Conference 1993.

[20] U. Shumlak and N. F. Roderick, Phys. Plasmas **5,** 2384 (1998).

[21] U. Shumlak and T. R. Jarboe, Phys. Plasmas **6,** 4382 (1999).

[22] U. Shumlak and T. R. Jarboe, Phys. Plasmas **7,** 2959 (2000).

[23] U. Shumlak, To be submitted to J. Comp. Phys.